

Inquire Network and Job Status, Symbol Table

This subsection covers the following topics:

- Inquire the Current Status of an Active Job Network
- Inquire Status of Whole Active Network
- Inquire Status of all Jobs of an Active Network Individually
- Inquire Symbol Table Used

Inquire the Current Status of an Active Job Network

You can inquire the current status of an active job network or of a single active job by using the following statement in your Natural application:

```
CALLNAT 'NOPUST2N'
      P-FUNCTION P-RC P-OWNER P-NETWORK P-JOB P-RUN
      P-SYMBOL-TABLE P-JOB-ID-10 P-STATUS-TIME
```

Meaning of the parameters:

Parameter	Format	Usage	
P-FUNCTION	A01	in	Function code:
		A	Get next active run number (in numerical order) beginning with a starting run number.
		R	Get last run number (in chronological order).
		S	Inquire status.
		N	Inquire status of next job (alphabetically).
		W	Status of next job waiting for anything, or of next job in status ‡permanent errorŸ.
		Y	Get name of symbol table from master network or job definition.
P-RC	N03	out	Return code:
		0	Function ok.
		1	Network or job not found, or run number does not exist.
		20	Waiting for symbol prompting, etc.
		21	Waiting for activation.
		25	Waiting for prerequisite.
		26	Job is in HOLD.
		27	Waiting for next action, or waiting for start time.
		28	Waiting for condition.
		29	Waiting for resource.

			30	Waiting for operating system server (node).
			31	JCL to be loaded.
			32	Network activation error.
			33	Job activation error.
			34	Symbol replacement error.
			35	Any schedule extraction or activation error.
			36	Job submission.
			37	Job submission error.
			38	Job execution.
			39	Job execution error.
			41	Any job currently executing.
			42	End-of-job checking.
			43	End-of-job checking error.
			44	End-of-job actions.
			45	End-of-job actions error.
			65	(All jobs) terminated ok.
			66	(At least one job) terminated not ok.
			69	(At least one) permanent error.
			93	Waiting for deactivation.
			101	Invalid function code.
			102	Parameters missing.
			999	Status not defined.
P-OWNER	A10	in	Owner of network.	
P-NETWORK	A10	in	Job network	
P-JOB	A10	in	Job. If left blank, inquiry is for whole network.	
P-RUN	P13	in out	Function A : Starting run number (can be 0). Function S : Run number to be checked. Function A : Next active run number (in numerical order). Function R : Last run number of the network in chronological order.	
P-SYMBOL-TABLE	A10	out	Name of the defined symbol table. Function Y , network: standard symbol table. When status inquiry is for whole network, first symbol table found is returned.	
P-JOB-ID-10	A10	out	ID of the job (for single jobs only, and only if the job was already submitted).	
P-STATUS-TIME	A14	out	Timestamp when the current status of the job was set. Format: YYYYMMDDHHIISS.	

Inquire Status of Whole Active Network

Use function **S**. Leave the P-JOB parameter blank.

Inquire Status of all Jobs of an Active Network Individually

Use function **N**. Delete the P-JOB parameter. Then invoke this API in a REPEAT-loop until you get RC = 1 (network end).

Each call returns the status of a job. The name is contained in P-JOB. Do not change the content of P-JOB, because it is used as starting value for the next call.

Inquire Symbol Table Used

The symbol table used is always returned for active networks and jobs.

For master networks and jobs, you can use the function **Y**.

Notes:

1. The status inquiry functions independently of Monitor activity, because it uses data base entries.
2. For a whole network inquiry (with function **S**), the status of the individual jobs are linked with logical AND. In the worst case, if at least one job has failed, the status **failed** is returned for the whole network.
3. An active network is considered to have **terminated ok** only if all its active jobs have terminated ok.
4. The status inquiry is only possible as long as the network or job involved has not been deactivated.
5. To obtain all active run numbers of a network, proceed as follows:
 - use function **A**
 - begin with starting run number 0
 - call the API
 - terminate when RC does not equal 0
 - keep result as next starting run number and continue with 3.